NIEM 3.0 Beta Tech Talk Transcript

Justin Stekervitz:

On behalf of the NIEM Program Management Office, welcome to the NIEM 3.0 Beta Tech Talk. My name is Justin Stekervitz and I am the Acting Managing Director of the NIEM PMO and also the NIEM Technical Architecture Committee Federal Co-Chair. Joining me on today's Tech Talk are two of our NIEM lead developers from the Georgia Tech Research Institute Mr. Mark Kindle and Mr. Webb Roberts.

If you are a frequent visitor to **www.NIEM.gov**, you should be aware that the development of NIEM 3.0 is currently underway. Today, we are in the Beta phase. As we get closer to the finalization of NIEM 3.0, it is critical that you, our community, are involved in the Beta review process to ensure a high-quality NIEM 3.0 release.

Thank you for your participation in today's Tech Talk and for your continued interest in NIEM 3.0. On today's call, we are going to explain what NIEM 3.0 is from a version perspective, we're going to provide an overview of the NIEM 3.0 release cycle and provide highlights from the last three months, and more importantly, focusing on the NIEM Beta release process itself. We're also going to explain what you can expect to find in the NIEM 3.0 Beta from an architectural standpoint. And lastly, and most importantly, how you as a member of our designated community can contribute in the review and make NIEM 3.0 the best product possible.

Moving into NIEM 3.0 and what does it mean? NIEM 3.0 is a major release, which means the Core part of our model is undergoing enhancements in terms of both architectural enhancements and content enhancements. During the major release, domains are also affected and architectural changes to the Core also impact the domains, as you will shortly see. Also during a major release, the domains have an opportunity to include new content or to modify existing content. A minor release reconciles the main updates to resolve conflicts created by new and modified domain content. It also allows domains to perform incremental harmonization on their schema set. NIEM 2.1 was considered a minor release, as it involved limited changes to a number of the NIEM domains.

The NIEM release history has been a careful balance of community needs, and resources available to develop a NIEM release, whether major or minor. As described on the previous slide a NIEM major release involves changes to NIEM Core and to the Domains. As you'll see in this slide, the last major release, NIEM 2.0 included versioning from a 1.0 Core to a 2.0 Core as well as bringing the domains up to the same version. For minor releases such as NIEM 2.1, NIEM Core does not change; however, the domains do. As such, the NIEM Core version is constant while the domain version is the next minor release duration, as an example we have NIEM 2.1. If you're familiar with NIEM 2.1, you would have seen new domains added, which include the Maritime or Chem/Bio/Rad/Nuc, and our Family Services domain. Existing domains were also modified to add new elements, or update existing elements such as necessary.

So why a NIEM 3.0? Well, as you're aware, NIEM is a community program, and as such, the NIEM data model is driven by the needs of the community. NIEM 3.0 addresses the needs that have been identified since NIEM Version 2.0, which was the last major release. Overall, NIEM 3.0 was informed by three important factors. Namely technical challenges experienced by NIEM domains, developers, and implementers; organizational challenges in adopting NIEM and new domains with requirements not previously met by a previous version of NIEM. So without further delay, I will turn the presentation over to Mr. Mark Kindl to walk through the NIEM 3.0 Release Cycle and get to the meat of the conversation. Mark?

Mark Kindl:

Thanks Justin. We'll move to the next slide here. First, I'm going to let you in a little bit on how these release cycles work. We've been through a pre-Alpha phase which you see on the far left. Pre-Alpha there's a lot of activity and that's when domains are preparing changes for content, and they're also resolving issues in the NIEM configuration control tool and generally a lot of activity. There's no release yet.

When we get to Alpha 1.0, that's when we have the first intermediate release. That's usually full of errors that have to be corrected through the Alpha phase and we've had an Alpha 1.0 and an Alpha 2.0 release. You've had a chance to see the Alpha 2.0 release, which also had some errors in it and now the Beta release is finally on the street. There is a URL to it where you can download it; it's not on this slide but I believe it's on one of the slides in the back. When we get to the Beta phase, the intermediate phase is starting to stabilize to some degree. I will warn you that there are still few errors in Beta, not too many. There shouldn't be too many. When we move to release candidate, there should be almost no errors at all and those that we find will be very minor refinements or corrections, leading into the final operational release.

On the previous slide you might have seen a little note saying that funneling begins with Beta 1.0. Funneling is a process that we apply in the Beta stage whereby anything reviewed that is passed during a review will not be returned to. If a subjective sort of error, someone say that a particular data component needs to be refactored or isn't quite right, if the NTAC or NBAC determine that is a subjective comment, once everything passes through that gate, we will not return to those things.

We will only return to things that have not been commented on yet during the Beta phase. So what that really says is that this is the time to make absolutely certain that the review is thorough. One exception to funneling is actual errors. If there is an error, outright error, then that will be corrected regardless of how many times it has been seen. You notice the comment at the bottom of this slide: "Silence is consent." If you really have a concern about something, please make that concern known, regardless of what that might take to fix, that decision will be made in the NTAC and NBAC Committee meetings.

Just to give you an idea of what goes on, this is a focused slide on the Beta timeline. Basically for about three weeks, the governance body that is the NTAC and NBAC vet and resolve the Alpha 2.0 issues and feedback. Then for the next three and half weeks in the middle, the developer goes through and makes the Beta release based on that feedback and the resolutions.

We are at the stage there in the sixth week where we're in the Beta review. Now we're in a two-week review period starting basically today. Feedback is due on the 8th of July. So if you are one of the community out there, **www.NIEM.gov** has a form for submitting feedback on Beta 1.0 and we highly encourage you to use it.

I'm going to give a little bit of an overview here on what you should see in the release. I'm not going to take a lot of time on this because it's mostly just pictures. In the upper left-hand corner of this slide, you'll see when you open up the package if you download it at the URL, it's zipped up, and you'll get a single directory. If you drop into that directory, you'll see the NIEM release under that directory. There will be an oasis catalog along with that on the same level. You will get a spreadsheet of the entire Beta release. I'll show you what that looks like here in a moment. You'll get a spreadsheet of the code list, and you will also get a spreadsheet of the quality assurance checklist that has been run on the release. And, there will be failure in there on some cases. Hopefully, not that many. If you open up the NIEM sub-directory there on the far left, that will open up to the list in the middle and if you were to open up domains, it would open up to the list on the right. The remainder of these pictures is basically a continuous, completely open release. So that top-level directory has been opened and this is what you would see if you expanded everything. Notice that we've placed the code list-all the code list is in one sub-director so they're not scattered over as previous releases were. And, finally back through the domains here, finally some of the externals and the infrastructure that supports the release.

Basically if you don't know what an oasis catalog is, this is the beginning of what you'll see. Basically it just tells a tool that understands the oasis catalogs, where all the schemas reside inside that release package. The model spreadsheet: this is just one page of not necessarily the Beta 1.0, but one page of a model spreadsheet from a NIEM release. You'll see across the bottom there's a tab for every domain so you can see the content of each domain in quite a bit of detail. Not just the property, but the type of property, the definition, and the other fields that are actually stored in the tool. So there's a keyword field, an example content field, and there's usage information. You'll notice in this spreadsheet most of those fields are blank, because most of those fields are blank in the model. They're not heavily populated yet simply because we haven't received that many inputs at this point, something that probably needs work in the future.

This is just a page from the list of enumerations that you will also have or see in the package that I mentioned. Pretty simple to follow. And then finally, the QA Report. This is an example of the QA Report from one of the Alpha versions. There's lots and lots of failures in it. Hopefully, there won't be that many failures in the Beta 1.0 when you open it, but there will be a few. You can expect a few, which will be corrected before RC 1.0.

In this release, there were a lot of resolved issues. So this gives you a chart, or this is a chart, that runs from August 2012 through February; and this is where we were tracking all the issues that were resolved in the NIEM Continuation Control Tool, by the NBAC in particular. You can see almost 100 issues resolved, 92 issues out of 109. There are still 17 issues in there that are unresolved, and those issues will be resolved probably in later releases. In addition to these hundred, there were also at least a hundred or more, maybe about 120 issues that are now resolved in 3.0 that were left over from the previous 2.0 release. They've been kind of hanging around for some years waiting for the next release so those have been integrated into this release, NIEM 3.0.

This is just a quick summary of what's in 3.0 that has changed from 2.1, the last release. You'll notice there's a new domain Biometrics, which is named Biometrics. There used to be a NIST domain that had some biometrics from the NIST operation that used to go on for a few years I believe. But this is a full-fledged biometrics domains and it includes a lot more than the old one did. It's much better, it's been corrected. A lot of errors were in the last one. There are updates to a number of the domains, you see them here, including a change in Family Services. Family Services changed it's name to Children, Youth and Family Services. I already mentioned the resolved issues, so there's about 200 plus issues that were resolved and placed into 3.0. There were a number of significant updates to large objects and they are listed here, the ones that are most significant, including location, document, contact information and the rest.

There are a number of new code lists that were added this time. We now have religion from the HL 7 folks; Gen-C is in there, which is the new country code list, basically a subset of I believe ISO-3166. Census U.S. county codes are there because of the sunset of many of the fixed codes. And there are several others. There are also some code list updates. Literally every code list coming from NIEM 2.1 had some sort of update to apply, with very few exceptions. The most significant ones are listed here. The sunset lists are still there for legacy purposes. We brought them all up to date, their state when they were finally retired and we're putting them in 3.0 for convenience, just so they're there in case anyone needs to go back and use them. There is one item missing from this list. There are some updates to ISO-3166-1, which is the ISO country code list. I believe we still need to apply some updates to that, so that will be in RC-1. That is one place where we are lacking in this particular Beta1.0 release. Finally, there was a lot of harmonization performed among the various domains and as well as some of the domains with Core. What we have listed here are some of the most significant ones.

I'm going to turn the presentation over to Webb Roberts, our Chief Architect and he's going to cover the architectural changes.

Webb Roberts:

Hi, I'm Webb Roberts. I've been working very closely with the NTAC to find the NIEM 3.0 technical architecture. Just a few points: the technical updates that I'm going to talk about today are not absolutely final. We are still in the Beta 1.0 of the NIEM release. These are going to be final when the NIEM release goes final. We're looking for comments, positive and negative, about the technical architecture in these details. These comments are going to be vetted by the NIEM Technical Architecture Committee, the NTAC, and I'm GTRI's representative on the NTAC, helping work through all the technical details.

Our goal in 3.0 is to focus on simplicity and usability. There are lots of things in NIEM 2.0 that were more complicated than they had to be. We've tried to simplify those things as much as we can, while maintaining a lot of functionality that's beyond what XML schema does by itself; and to make these things more straightforward to use and more useful for the developers. The NIEM Naming Design Rules (NDR) is in the process of being updated and we're looking for comments on the NIEM Beta Release Process itself. These comments are going to influence the NDR as it's being developed. We have draft rules that have been applied to this Beta for the quality control, and those are going through a spiral process of being published as time passes. But right now, we're focusing on the Beta 1.0 schema release as the subject of comments.

So in 3.0, we've made a lot of updates to structures. Structures is the fundamental, foundational schema in NIEM that defines the base types and base elements for NIEM content schemas. We've re-factored a lot of this to simplify how this works. In NIEM 2.0 we had a link meta data attribute which was meta data on relationships between objects and we've renamed that to relationship meta data. The sequence id provided an alternative method of ordering of the natural content of XML instances. Since we've used XML schemas sequenced content model, it was hard to reorganize the order of things. So sequence id is the method for doing that. That's been stood down and you'll see there are several different ways of doing sequences in various types inside of NIEM. The augmentation element and the meta data element have been stood down. They've been removed from NIEM 3.0. They were abstract substitution heads for augmentations in meta data and those didn't serve a positive purpose. In meta data, they sort of complicated how things got processed so we've removed those.

The structures complex object type has been forked; it's been divided into two types, the object type and the association type. Whereas before, someone who was building an association would build off the complex object type as an extension base and apply an annotation that indicated it was an association. In NIEM 3.0, it will merely extend from the structure's association type. As a result, the resources of object and association that occurred in structures have been removed. We've also removed the structure's reference type. This was a result of modifying the way we do reference elements from NIEM 2.0 to NIEM 3.0. We have merged content and reference elements so that you don't need to define reference elements twice and reference types are no longer necessary.

NIEM has always had a basic set of annotations that were applied to the name schemas and the appinfo namespace. The appinfo namespace has been heavily re-factored between 2.0 and 3.0 and has been greatly simplified in NIEM 3.0, and hopefully you'll be pleased by the changes. The main changes that happened in the annotations, whereas in NIEM 2.0 we entirely used element annotations, meaning XML elements that appeared as children of appinfo elements inside component definitions. In NIEM 3.0 those have been uniformly replaced with attributes. So the machine-readable annotations in NIEM 3.0 are entirely attributes that appear on the XML schema components. The conformant indicator has always served two purposes. The first was to indicate that a schema is itself intended to be a NIEM-conformant schema. That use has been replaced with an attribute called conformance targets, which has a value of a list of URIs, and each URI represent a conformance target to which the schema is intended to apply. The other purpose of the conformance indicator was to indicate that an XML schema import is an import of a non-conformance schema. It was an annotation of an XML schema that was intended to indicate that it was not a conformance schema. And that's been replaced by an external import indicator which has a value of true if it is an XML import schema.

The appinfo base has been totally stood down. This was responsible for indicating whether something was an object or association and it was designed initially with the intention of supporting multiple inheritances, and that has been stood down completely. The annotation of reference target was part of the previous reference mechanism that went hand in hand with reference type and that's been taken out. Instead, the reference target of an element is identified by the XML schema's type attribute on the element. So an element of type, person type will have a reference target of person type so we don't have to duplicate that with a separate reference target annotation. The element applies-to always had two purposes. One was augmentation, indicating to what types an augmentation might be applied and also to indicate what element or types of meta data element might be applied. These have been refactored; the augmentation methodology has been completely re-done. The concept of augmentations is still there but instead of an annotation, it's being identified by a substitution group for augmentation element.

For the meta data, the applies-to is still a meaningful concept but it has been split into two attributes, one of applies-to types and one of applies-to elements. The first indicating to what type a meta data element might apply and the other indicating to what elements a meta data element might apply. Last, the external adaptor type indicator has been refactored to an attribute. So, external adaptor type has the attribute external adaptor type indicator set to true.

So here on slide 25, we show the example of an annotation simplified for a fairly straightforward complex type. Here we actually have an external adaptor for a multi-curve, defined by GML and we see that there is a new attribute. So the things in green are new content and the things in red are things that were in NIEM 2.0 and have been removed. We'll see that the external adaptor type indicator is now an attribute where as before it was an element content of appinfo. And, we also see that the whole appinfo block has been removed. The base is no longer necessary and the external adaptor type indicator has been promoted up to an attribute. We also see that the base type is now object type instead of complex object type which removes the necessity of that annotation.

On slide 26, we see the unification of content and reference cells. Before NIEM 2.0, when we wanted a reference element, we would define an element such as weapon user reference. And when we wanted a content element, we would define a content like weapon user. These have been merged into a single element. So we see on the left, we have an arrest with an involved weapon, which has a weapon user, and weapon user is a person. And sometimes you'll want that to be in content, meaning a weapon user has a person birthdate and a person name. But sometimes you'll want that to be a reference as a person that is identified elsewhere in the message. So instead of having two elements now, the weapon user would have a reference attribute, structures: ref, that has an id value and that id is the id of an element type person type somewhere else in this XML instance. So in this case, we see, "Person 1" refers to a name core person element with structures id Person 1.

The refactoring here shows the simplified schema in defining the content versus reference. We see here weapon type with the appinfo block no longer necessary in NIEM 3.0. Object type instead of complex object type, and whereas before, in order to represent a flexible content model for an object, you would have had to have a weapon user and a weapon user reference element. In NIEM 3.0, that weapon user reference element is not needed. It needs to be referenced by weapon type in order to find at the top left. So in NIEM 3.0 we have a weapon user of type person type. We've cut out a lot of unnecessary schema definitions.

In NIEM 3.0 we've updated the augmentations methodology. Augmentations are a way that a namespace, in a particular domain, can define an extension to a base type without needing sub-classes. So this example is a NIEM Core person type for which Immigration and Intel need to have additional information they convey about the person. Now what they don't need to do in this case is to find a new special type of person that is an extension of a base person type. Instead, what they're saying is these are additional characteristics of a person that are specific to our domain. And the method for doing this now is a type, such as the person type defines a person augmentation point. The name of the element is constrained to be the same as the name of the type but with the word type being replaced by augmentation point, which is an element that has no type and is set to be abstract. Being typeless means that you can substitute elements of various types for it. Being abstract means that the element itself is not allowed to appear in an instance so it can't act as a wildcard. Any domain that needs to find an augmentation for example, the SIFUS person augmentation defines that element of an augmentation type and sets it to be substitutable for the augmentation point. And this way, the schema says we have a person type and additional information about a person is defined by the SIFUS person augmentation with the listed content.

We've refactored the structures base types. This just shows all the changes we talked about and rolled into it. Complex object type has been renamed; it contains an augmentation point which is defined as structures augmentation point, which has the same qualified name as the type but with the word type replaced with the word augmentation point. So you could substitute in for object augmentation point if you wanted to define something that has said additional stuff about any given object. We also added the structures: ref which provides the reference element or the flexible

content of content or reference. We've renamed link meta data to relationship meta data, and last we see the addition of a wildcard for IC ISM and NTK namespaces. These are defined by the Intel community as a trusted data format which includes the ISM and NTK. ISM is for security markings and NTK is for need-to-know. If you need to work inside a TDF environment, this should provide all that you need to support the need markup inside payloads of TDF messages. We'll provide details elsewhere on how that should work.

In addition for NIEM 3.0, we've added a local vocabulary. Local vocabulary is jargon, acronyms, and slang that are used within a particular domain or community of interest. In NIEM 2.0, the rule was that words used in the names of elements or attributes or types had to be dictionary terms, with the exception of things occurring in naming and design rules and abbreviations. This has been stood down for 3.0. Instead, the rule is that component names need to be dictionary terms with the exception of things that are identified as explicitly local vocabulary in the namespace. And, we see this in the NIEM 3.0 Beta 1.0 in the Immigration namespace. We have ICE and USCIS defined as terms that are going to be used in component names insides the namespace. That way when a user or developer comes in and looks and sees ICE employee, they know where to look for the definition of the word ICE and can find exactly what that terms means inside that namespace. So as a result, we've removed the table of abbreviations from the NDR. It's not going to show up in the 3.0 NDR. And, domains are encouraged to use their best local jargon that most accurately reflects the terms inside their namespace.

We've also refactored the way we do codes and text inside the NIEM 3.0 release. This isn't really a structures or appinfo change, this is just a change to the standard practice when we define NIEM Core. Before in NIEM 2.0 for a situation when we had a code list we would define as an aircraft type would have an aircraft fuselage color as a thing that it needed. We would define an aircraft fuselage color code and aircraft fuselage color text as sub-element of aircraft type and we've changed that to where the aircraft fuselage color is an abstract typeless element that acts as a base element and the code and the text representations are substitutable for that base element. We have an aircraft type with a fuselage color that's an abstract, and then the color code and the color text are defined as substitutions. And, this enables anybody who's creating an IEPD, or an extension, or upgrading a domain, or upgrading a code list to have a hook where they can substitute new representations without the losing the agreed upon meaning defined by NIEM Core of what an aircraft fuselage color is in this case.

So I'm going to pass it back Justin and he'll describe how to submit comments.

Justin Stekervitz:

Thank you Webb. So now onto how you can help us. So for the Beta Review process, slide 33 provides a great deal of information. We're looking for feedback from now until July 8th. The Beta model can be downloaded from our website. The URL is included on this slide. And it's important to remember that Beta is your last chance for you to provide your feedback on the entire NIEM 3.0 release. Mark discussed funneling and that's important to keep in mind as you're doing your reviews. The NIEM PMO, committees, and our partners have worked hard to produce NIEM 3.0. But, in order to ensure that NIEM 3.0 is a success, and by that we mean a quality product, we need your assistance in reviewing the Beta Release.

So this particular slide, use this as your instruction guide. Step 1—this slide just gives a little bit more detail in terms of where to go to download the Beta model and things to look for. And then depending on the type of user, if you're doing a non-technical review, this provides an Excel file for you to understand what has changed in NIEM from a Core standpoint and a NIEM domain standpoint. And from the technical sense, we have the tooling available, the SSGT or subset schema generator, which will allow you to search the model to perform your review.

Slide 36 discusses some of the things you want to look for and to make sure that the definitions are accurate, that grammar, spelling and punctuation errors have been taken care of. Comments on our technical architecture, harmonization, and code lists are very important. And any other issues you may see as you are performing your review.

And then lastly, how to go about submitting your feedback—**www.NIEM.gov** has a feedback page available and the link is noted in this slide. The key thing to keep in mind is to ensure that you select NIEM 3.0 feedback as your category when you are submitting your comments, that way your comments will make it to the top and get to the appropriate people for review.

Don't miss your last chance! We are going public for comments and we expect to receive a number of them. Again, the submission period is from today, June 25th until July 8th, which is a little less than two weeks. The link to download the model is included on the slide as is the link to submit feedback. The NIEM PMO is a resource and feel free to email us any questions with the subject "3.0 Help" and we'll get back to you as soon as possible to ensure that you can provide your feedback.

On behalf of the NIEM PMO, I want to express my appreciation for your continued support and interest in the program. I hope you found the information presented today a benefit and I think now we are going to take some questions.

Jenny Yi:

Thank you Justin and before we move on to the specific questions, I just wanted to let everyone know that the recording for this presentation will be made available at **www.NIEM.gov/version3.0**. You can see this link in the important box. We will also make this slide presentation available as well in the same location, so you'll be able to get more details at the link there.

So let's get started on some of the detailed questions. So a question we got from an email was a person is interested in developing a community and developing a data model. They're wondering what domains they should leverage and what information they should leverage to start that activity?

Justin Stekervitz:

We have a numbers of resources on your website, but probably the first step is contact **information@niem.gov** or to use our online help desk and we can point you in the right direction and get you connected with the appropriate people.

Jenny Yi:

The next question is "Will NIEM 3.0 Core be larger or smaller than NIEM 2.0 Core?"

Mark Kindl:

It should be somewhat smaller because at the NBAC's request, we have removed some of the material in NIEM Core and moved it into appropriate domain, so it is constraining now the size of Core. Also, if you consider that the reference content and inline content have been merged into a single element, that of course reduces the size of Core somewhat, but it is not because you've reduced the number of concepts it's just that you've reduced the overhead for them in that case. But we have moved some of the elements, properties, and some types out of Core and into some of the domains so that it is a little bit smaller. But I wouldn't be looking at it for being incredibly small. It's probably still a little bloated at this point.

Jenny Yi:

Our next question is "Will the NIEM 3.0 Beta Release include documentation, in particular an updated NDR?"

Webb Roberts:

So the NDR is not part of the Beta 1.0 release. We're looking forward to getting the NDR through an NTAC review process. Right now, it's in a very draft form where the rules are being constructed and it's going to be subject to review before we get it out for the final release.

Jenny Yi:

Our next question is "Will there be a UML version of any part of NIEM 3.0?"

Justin Stekervitz:

We are working on a NIEM UML profile for NIEM 3.0 that would be similar to the one that is currently in Beta through the Object Management Group. We expect that would come out approximately three months following the release of NIEM 3.0 just so that we could maintain alignment to the Object Management Group's schedule for bringing forth specifications for finalization and adoption. Mark, do you want to provide any context with regards to what you're doing for UML?

Mark Kindl:

At one point, we associated with the tools, we were working toward at least a preliminary output of a subset for example as a UML model. I really don't want to say at this point how far we are going to get with that.

Jenny Yi:

The next question is, "Is NIEM data model easier to construct in version 3.0?"

Webb Roberts:

I think from a technical architecture perspective, as far as the difficulty of creating a schema that is NIEM conformant, I think it is going to be a good bit simpler. We still use the Garden of Eden content model with top level elements and top level types, but the requirements for annotations and bases, the number of things you could possibly get wrong is a good bit smaller. It's a bit simpler, it's a little bit more straight-forward to design NIEM conformant schemas under 3.0 than it was under 2.0.

Jenny Yi:

Our next question is, "The comparison between attributes and elements: is there a reasoning behind the change, and a preference for using attributes over elements?"

Webb Roberts:

Absolutely. The change has been in XML schema annotations, which previously were done as appinfo elements and are now being done as machine readable annotations. This change really came down from the DOD side and efforts from C2Core and later MilOps to make things a little bit more straight-forward and support various kinds of machine readable annotations. The concept before with appinfo elements has been generalized into machine-readable elements which includes appinfo elements as well as attributes from namespaces other than the XML schema namespaces that are applied to the XML schema components.

Looking at 3.0, we had the opportunity to make those schemas as simple as they could be, so where we could adopt attributes instead of elements, we did it, and that greatly reduces the amount of overhead to define and use content and annotations inside of schemas. So the fundamental reason was that attributes cover the need very, very well; it's easier to build schemas with attributes than elements; it's easier to define the content with attributes than elements; and it really simplifies the process of writing and validating schemas.

Jenny Yi:

That actually leads into a really good segway into the next question, which is, "Has the DOD UCore been added as a domain?"

Justin Stekervitz:

They are working to develop a MilOps domain. Further details would be distributed through the Department of Defense.

Webb Roberts:

I want to underscore that I think the NIEM 3.0 support for the trusted data format shows the commitment to support the needs of DOD. TDF really covers some of the requirements of UCore and supports the needs of the Intel community.

Jenny Yi:

The next question is, "The complex objects are used more than projected associations, but if the nature of the object is complex, what is a new way to describe it?"

Webb Roberts:

Before we had complex object type as the base type for complex types with complex content inside of NIEM; and that's been forked to an object type which represents an object, a thing, and an association type, which represents a relationship between things. So before in NIEM 2.0, you would use complex object type as a base, now you use either object or association depending on what you're representing. As well, we still have the attribute group that carries in needed attributes into complex types with simple content so when you're building complex types with simple content, you incorporate that content group in the object you're building. I think it's a lot like it was NIEM 2.0, it's a little bit more straight-forward because you don't need annotations of base associations, instead their base types, but otherwise it's very similar to 2.0.

Jenny Yi:

We have a question if, "Will comments submitted last week be in Beta or do they need to be in this time window?"

Mark Kindl:

We still have feedback that we're working on so if you've already submitted comments, they're still there. The answer to that is no: there's no reason to re-submit comments.

Justin Stekervitz:

If I could also add to that, I think this also gets to another question. For those folks who provided comments on Alpha 2.0 and we didn't reached out to you directly, we will be doing so in the coming week to respond to the comments or questions that you put forth.

Jenny Yi:

The next question is, "Is there a difference log between Alpha 2.0 and Beta 1.0 that is available for review?"

Mark Kindl:

There's a high-level change log at this time but normally we don't put out the detailed change log until we get to RC-1.

Jenny Yi:

The last question I have is, "Are there any significant updates to the SSGT and the Want List generation?"

Mark Kindl:

They are being updated to adjust to the 3.0 architecture, and most of those changes have been made at this point. I'm not sure how this relates to the Want List. The Want List is essentially just a format for saving the subset.

Webb Roberts:

I don't think we have significant changes to the Want List format. I think there are some Want List values that won't be useful in 3.0 because of the way we're doing content versus reference but otherwise it stays the same as the 2.0 Want List.

Mark Kindl:

So in other words, the schema that defines the Want List is not expected to change. If it does, we certainly will post an updated schema. But at this point, that is not expected to change.

Jenny Yi:

I know that we're going to close to the end. If there were any questions that were not answered, please email us at **information@niem.gov**. Also, I know folks are asking for the PDF and online recording. They have not yet been made available but they will be made available a little later. The recording will be made a little bit later, and you can find that at **www.niem.gov/version3**. That URL is also in the important box on the right hand side.

At this time, I just want to thank the presenters and participants for taking the time out of their days to discuss this. Again, if you have any questions, please send them to **information@niem.gov**. If you'd like to submit feedback, and we hope that you do, please visit **www.niem.gov/Pages/contact.aspx** and you can also find additional information about 3.0 at **www.niem.gov/version3**.

We hope to see you online and thank you again for participating.